Power Expectation Propagation for the Relevance Tagging Machine

Dmitry Molchanov
Dmitry Kondrashkin
Dmitry Vetrov

November 06, 2015

Complex distributions

$$p(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$$

$$p(\mathbf{X}) = \int p(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) d\mathbf{Y} d\mathbf{Z} = ?$$

$$\mathbb{E}_{p(\mathbf{X})} f(\mathbf{X}) = \int f(\mathbf{X}) p(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) d\mathbf{X} d\mathbf{Y} d\mathbf{Z} = ?$$

Distribution approximation

Consider a distribution $p(x)$ from a complex distribution family. The goal is to approximate it with a distribution $q(x)$ from a simple distribution family:

$$p(x) \approx q(x)$$

$$p(x) = \prod_{i=1}^{n} t_i(x), \;\; t_i \in \mathcal{F}_i \text{ – complex family of distributions}$$

$$q(x) = \prod_{i=1}^{n} \tilde{t}_i(x), \;\; \tilde{t}_i \in \mathcal{Q}_i \text{ – simple family of distributions}$$

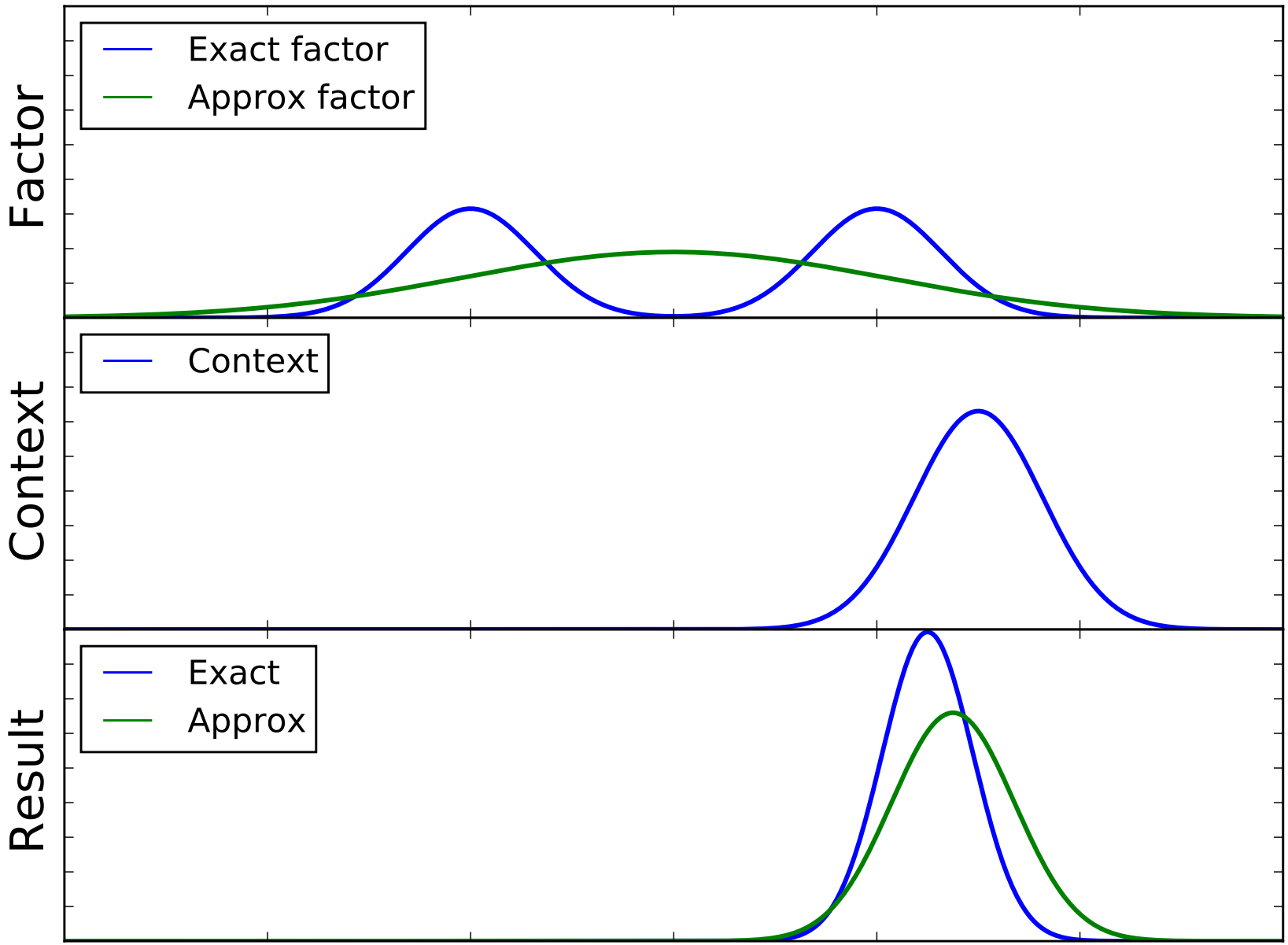
Distribution approximation: naive approach

$$p(x) = \prod_{i=1}^{n} t_i(x), \qquad q(x) = \prod_{i=1}^{n} \tilde{t}_i(x)$$

$$\tilde{t}_i(x) \approx t_i(x)?$$

Bad idea!

Distribution approximation: naive approach



Contextual approximation

$$p(x) = \prod_{i=1}^{n} t_i(x), \quad q(x) = \prod_{i=1}^{n} \tilde{t}_i(x)$$

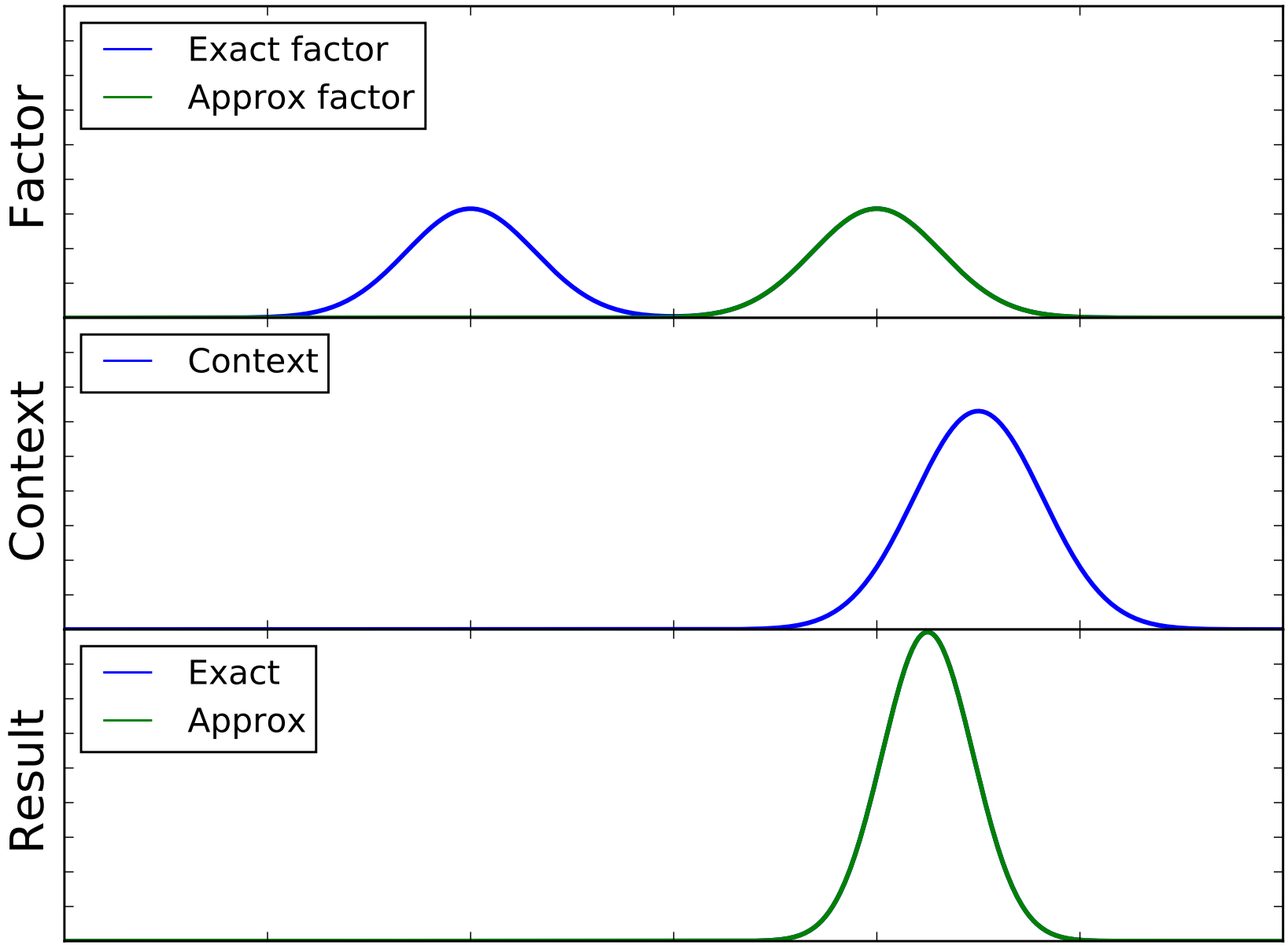
Context for i-th factor:

$$q^{\setminus i}(x) = \frac{q(x)}{\tilde{t}_i(x)}$$

Contextual approximation:

$$q^{\setminus i}(x)\tilde{t}_i(x) \approx q^{\setminus i}(x)t_i(x)$$

Contextual approximation



KL-divergence

A brief reminder: Kullback–Leibler divergence

$$\mathrm{KL}(p\|q) = \int p(x) \log \frac{p(x)}{q(x)} dx$$

Some properties:

- $\mathrm{KL}(p\|q) \geq 0$
- $\mathrm{KL}(p\|q) = 0 \Leftrightarrow p(x) = q(x)$ almost everywhere
- Can be used to measure the difference between p and q, ...
- ... but is not a metric and is not symmetric

KL-projection

Contextual approximation:

$$q^{\setminus i}(x)\tilde{t}_i(x) \approx q^{\setminus i}(x)t_i(x)$$

KL-projection:

$$q^{\setminus i}(x)\tilde{t}_i(x) = \arg\min_{g\in\mathcal{Q}} \mathrm{KL}(q^{\setminus i}\cdot t_i \mid\mid g) = \mathrm{proj}_{\mathcal{Q}}[q^{\setminus i}\cdot t_i]$$

KL-projection in exponential family

Let $\mathcal{Q}$ be a subset of exponential family with parameters $\boldsymbol{\theta}$:

$$q(\boldsymbol{x} \mid \boldsymbol{\theta}) = \frac{g(\boldsymbol{x})}{h(\boldsymbol{\theta})} \exp(\boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x}))$$

$\boldsymbol{\phi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \ldots, \phi_m(\boldsymbol{x}))^T$ are sufficient statistics of $\mathcal{Q}$.

In this case KL-projection on $\mathcal{Q}$ is equivalent to moment matching:

$$g = \mathrm{proj}_{\mathcal{Q}}[f]$$

$$\Updownarrow$$

$$\int \phi_j(\boldsymbol{x}) g(\boldsymbol{x}) d\boldsymbol{x} = \int \phi_j(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x}, \ \forall j = 1..m$$

EP algorithm

1. Initialize $q = \prod_{i=1}^{n} \tilde{t}_i$
2. Recalculate context for all factors: $q^{\setminus i}(x) = q(x)/\tilde{t}_i(x)$
3. Update $\tilde{t}_i = \mathrm{proj}_{\mathcal{Q}}[q^{\setminus i} \cdot t_i]/q^{\setminus i}$
4. Repeat steps 2-3 until convergence

\+ Good in practice: fast and accurate

– Poor theoretical evidence to its convergence or approximation accuracy

– Fails if moment-matching in proj operator is intractable

Power EP motivation

EP fails if $\int \phi_j(x) q^{\backslash i}(x) t_i(x) dx$ is intractable.

However, $\int \phi_j(x) q^{\backslash i}(x) (t_i(x))^{\eta_i} dx$ may be tractable for some η_i.

For example:

$$q(x) \text{ is a Gaussian, } \phi_1(x) = x, \ \phi_2(x) = x^2$$

$$t_i(x) \text{ is Student's } t\text{-distribution, } t_i(x) = \frac{1}{x^2 + 1}$$

$$\int x^k \frac{1}{x^2+1} \mathcal{N}(x \,|\, \mu, \sigma^2) dx = ?$$

$$\int x^k \left(\frac{1}{x^2+1} \right)^{-1} \mathcal{N}(x \,|\, \mu, \sigma^2) dx = \int x^k (x^2+1) \mathcal{N}(x \,|\, \mu, \sigma^2) dx$$

Power EP

$$\text{EP:} \qquad \frac{q(x)}{\tilde{t}_i(x)}\tilde{t}_i(x) \approx \frac{q(x)}{\tilde{t}_i(x)}t_i(x)$$

$$\text{Power EP:} \quad \frac{q(x)}{\left(\tilde{t}_i(x)\right)^{\eta_i}}\left(\tilde{t}_i(x)\right)^{\eta_i} \approx \frac{q(x)}{\left(\tilde{t}_i(x)\right)^{\eta_i}}\left(t_i(x)\right)^{\eta_i}$$

$$q^{\setminus i}(x) = \frac{q(x)}{\left(\tilde{t}_i(x)\right)^{\eta_i}}$$

$$\tilde{t}_i = \left(\frac{\text{proj}_{\mathcal{Q}}[q^{\setminus i} \cdot t_i^{\eta_i}]}{q^{\setminus i}}\right)^{\frac{1}{\eta_i}}$$

\+ Is applicable to a wider variety of models

Relevance Tagging Machine

We address the binary classification problem with binary features.

- $(\boldsymbol{x}_i, c_i)_{i=1}^n$ is the training set
- $\boldsymbol{x}_i \in \{0,1\}^d$ is an object and $c_i \in \{0,1\}$ is it's class label
- $x_{ij} = 1 \Leftrightarrow \boldsymbol{x}_i$ is tagged by the tag j
- All tags affect the class label independently

Probabilistic model of the RTM:

$$\theta_j = \mathrm{P}(c = 1 \,|\, x_j = 1), \qquad \mathrm{P}(c = 1 \,|\, \boldsymbol{x}, \boldsymbol{\theta}) = \frac{\prod\limits_{j=1}^{d} \theta_j^{x_j}}{\prod\limits_{j=1}^{d} \theta_j^{x_j} + \prod\limits_{j=1}^{d} (1-\theta_j)^{x_j}}$$

ARD

Bayesian *automatic relevance determination* (ARD) approach.

- Parameters are given independent priors
- Hyperparameters are trained by maximizing the evidence

Symmetrical Beta distribution:

$$\theta_j \sim \mathrm{Beta}(\theta_j \,|\, \alpha_j, \alpha_j),\ \alpha_j \in [1, +\infty)$$

- $\alpha_j = 1 \Rightarrow \theta_j^{MAP} = \theta_j^{ML}$
- $\alpha_j = +\infty \Rightarrow \theta_j = 0.5 \Rightarrow \theta_j$ is removed from the model

Evidence maximization:

$$\int \mathrm{P}(\boldsymbol{c} \,|\, \mathbf{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta} \,|\, \boldsymbol{\alpha}) d\boldsymbol{\theta} \to \max_{\boldsymbol{\alpha}}$$

Likelihood approximation

$$\mathrm{P}(\boldsymbol{c} \,|\, \mathbf{X}, \boldsymbol{\theta}) = \prod_{i=1}^{n} \mathrm{P}(c_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}) = \prod_{i=1}^{n} t_i(\boldsymbol{\theta}) \approx \frac{1}{Z} q(\boldsymbol{\theta})$$

$$q(\boldsymbol{\theta}) = \prod_{i=1}^{n} \tilde{t}_i(\boldsymbol{\theta}) = \prod_{i=1}^{n} \prod_{j=1}^{d} \mathrm{Beta}(\theta_j \,|\, a_{ij}, b_{ij})$$

$$t_i(\boldsymbol{\theta}) = \frac{\prod_{j=1}^{d} \theta_j^{c_i x_{ij}} (1-\theta_j)^{(1-c_i)x_{ij}}}{\prod_{j=1}^{d} \theta_j^{x_{ij}} + \prod_{j=1}^{d} (1-\theta_j)^{x_{ij}}}$$

Likelihood approximation

$$t_i(\boldsymbol{\theta})^{-1} = \left(\frac{\prod_{j=1}^{d} \theta_j^{c_i x_{ij}} (1-\theta_j)^{(1-c_i)x_{ij}}}{\prod_{j=1}^{d} \theta_j^{x_{ij}} + \prod_{j=1}^{d} (1-\theta_j)^{x_{ij}}} \right)^{-1} =$$

$$= \prod_{j=1}^{d} \theta_j^{x_{ij}(1-c_i)} (1-\theta_j)^{x_{ij}(c_i-1)} +$$

$$+ \prod_{j=1}^{d} \theta_j^{-x_{ij}c_i} (1-\theta_j)^{x_{ij}c_i}$$

Power EP update for the RTM

$$\tilde{t}_i(\boldsymbol{\theta})^{new} = \left(\frac{\mathrm{proj}[t_i(\boldsymbol{\theta})^{-1} \cdot q^{\setminus i}(\boldsymbol{\theta})]}{q^{\setminus i}(\boldsymbol{\theta})} \right)^{-1}$$

$$t_i(\boldsymbol{\theta})^{-1} \cdot q^{\setminus i}(\boldsymbol{\theta}) \propto \ldots$$

$$\cdots \propto \prod_{j=1}^{d} \theta_j^{A_{ij}^1 - 1} (1-\theta_j)^{B_{ij}^1 - 1} + \prod_{j=1}^{d} \theta_j^{A_{ij}^2 - 1} (1-\theta_j)^{B_{ij}^2 - 1}$$

Sufficient statistics of beta distribution $p(x) = \mathrm{Beta}(x \,|\, a, b)$:
$\boldsymbol{\phi}(x) = (\log x,\ \log(1-x))^T$
Their expectation ($\psi(x)$ is the digamma function):

$$\mathbb{E}_{p(x)} \log x = \psi(a) - \psi(a+b),$$

$$\mathbb{E}_{p(x)} \log(1-x) = \psi(b) - \psi(a+b)$$

Power EP update for the RTM

Moment matching:

$$\mathrm{proj}[t_i(\boldsymbol{\theta})^{-1} \cdot q^{\setminus i}(\boldsymbol{\theta})] = \prod_{j=1}^{d} \mathrm{Beta}(\theta_j \mid \tilde{A}_{ij}, \tilde{B}_{ij})$$

$$\begin{cases} \psi(\tilde{A}_{ij}) - \psi(\tilde{A}_{ij} + \tilde{B}_{ij}) = c_1 \\ \psi(\tilde{B}_{ij}) - \psi(\tilde{A}_{ij} + \tilde{B}_{ij}) = c_2 \end{cases}$$

It can be solved efficiently with 3–5 iterations of Newton's method [Minka, Estimating a Dirichlet distribution, 2012].

Convergence tricks

Power EP algorithm for the RTM:

$$q^{\setminus i}(x) = q(x) \cdot \tilde{t}_i(x)$$

$$\tilde{t}_i = \left(\frac{\mathrm{proj}_{\mathcal{Q}}[q^{\setminus i} \cdot t_i^{-1}]}{q^{\setminus i}} \right)^{-1}$$

The algorithm, written in this form, does not converge.
We used several tricks to improve its stability:

1. Damping
2. Sequential update of factors
3. More damping
4. Addition of fixed prior distribution factors

Convergence tricks

Damping

for $k \leftarrow 1$ **to** *maximum iteration number or until convergence* **do**

$$\tilde{t}_i^{old} := \tilde{t}_i \quad \forall i \; q^{\setminus i}(x) := q^{new}(x) \cdot \tilde{t}_i(x) \quad \forall i$$

$$\tilde{t}_i^{new} := \left(\frac{\mathtt{proj}_{\mathcal{Q}}[q^{\setminus i} \cdot t_i^{-1}]}{q^{\setminus i}} \right)^{-1} \quad \forall i$$

$$\tilde{t}_i := \left(\tilde{t}_i^{old}\right)^{\gamma_i} \cdot \left(\tilde{t}_i^{new}\right)^{1-\gamma_i}, \quad \forall i, \; \gamma_i \in [0, 1)$$

end

Parameters of $\tilde{t}_i$ can be computed as a convex combination of old and new ones.

Convergence tricks

Sequential update of factors

for $k \leftarrow 1$ **to** *maximum iteration number or until convergence* **do**

  $\tilde{t}_i^{old} := \tilde{t}_i \quad \forall i$

  for $i \leftarrow 1$ **to** n **do**

    $$q^{new}(x) := \prod_{l=1}^{i-1} \tilde{t}_l^{new} \cdot \prod_{l=i}^{n} \tilde{t}_l$$

    $$q^{\setminus i}(x) := q^{new}(x) \cdot \tilde{t}_i(x)$$

    $$\tilde{t}_i^{new} := \left(\frac{\mathtt{proj}_{\mathcal{Q}}[q^{\setminus i} \cdot t_i^{-1}]}{q^{\setminus i}} \right)^{-1}$$

  end

  $\tilde{t}_i := \left(\tilde{t}_i^{old}\right)^{\gamma_i} \cdot \left(\tilde{t}_i^{new}\right)^{1-\gamma_i}, \quad \gamma_i \in [0, 1) \quad \forall i$

end

Convergence tricks

More damping

for $k \leftarrow 1$ **to** *maximum iteration number or until convergence* **do**

$$\tilde{t}_i^{old} := \tilde{t}_i \quad \forall i$$

for $i \leftarrow 1$ **to** n **do**

$$q^{new}(x) := \prod_{l=1}^{i-1} \tilde{t}_l^{new} \cdot \prod_{l=i}^{n} \tilde{t}_l$$

$$q^{\setminus i}(x) := q^{new}(x) \cdot \tilde{t}_i(x)$$

$$\tilde{t}_i^{new} := \tilde{t}_i^{\rho_i} \cdot \left(\left(\frac{\mathtt{proj}_{\mathcal{Q}}[q^{\setminus i} \cdot t_i^{-1}]}{q^{\setminus i}} \right)^{-1} \right)^{(1-\rho_i)}$$

end

$$\tilde{t}_i := \left(\tilde{t}_i^{old}\right)^{\gamma_i} \cdot \left(\tilde{t}_i^{new}\right)^{1-\gamma_i}, \quad \gamma_i \in [0,1) \quad \forall i$$

end

Convergence tricks

Addition of fixed prior distribution factors

$$p(\boldsymbol{\theta} \mid \boldsymbol{\alpha}^0) = \prod_{j=1}^{d} \mathrm{Beta}(\theta_j \mid \alpha_j^0, \alpha_j^0)$$

$$\mathrm{P}(\boldsymbol{c} \mid \mathbf{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \boldsymbol{\alpha}^0) = p(\boldsymbol{\theta} \mid \boldsymbol{\alpha}^0) \prod_{i=1}^{n} \mathrm{P}(c_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}) =$$

$$= p(\boldsymbol{\theta} \mid \boldsymbol{\alpha}^0) \prod_{i=1}^{n} t_i(\boldsymbol{\theta}) \approx \frac{1}{Z} q(\boldsymbol{\theta})$$

$$q(\boldsymbol{\theta}) = p(\boldsymbol{\theta} \mid \boldsymbol{\alpha}^0) \prod_{i=1}^{n} \tilde{t}_i(\boldsymbol{\theta}) =$$

$$= p(\boldsymbol{\theta} \mid \boldsymbol{\alpha}^0) \prod_{i=1}^{n} \prod_{j=1}^{d} \mathrm{Beta}(\theta_j \mid a_{ij}, b_{ij})$$

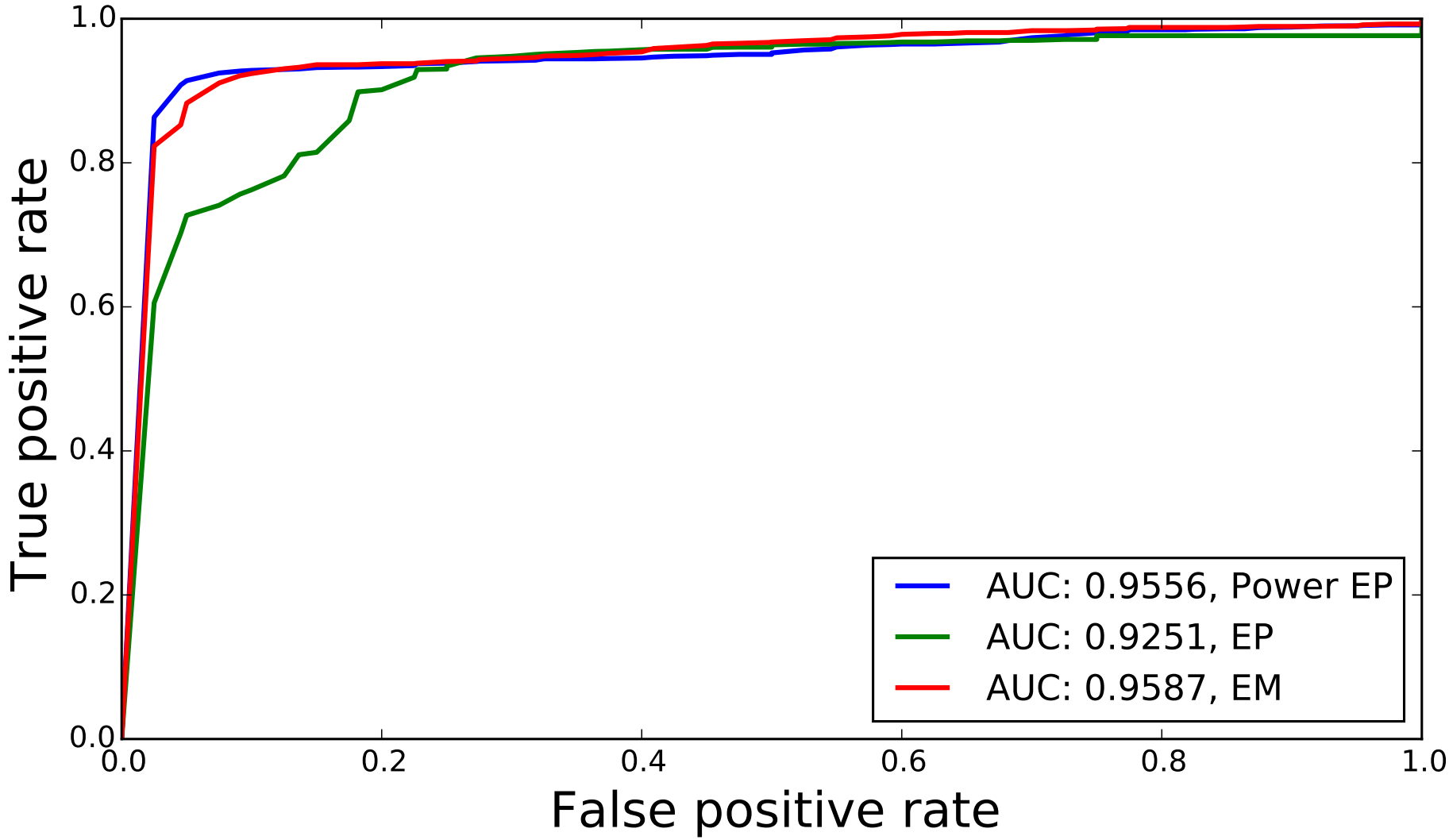
Other training methods for the RTM

- Approximate Expectation Propagation (EP)
- Coordinate-wise optimization of variational evidence lower bound (EM)

Experiments

Synthetic data

ROC curve for feature selection task:



Experiments

Sentiment analysis

Test set classification accuracy:

	Power EP	EM	EP
Dataset 1:	0.9708	0.9659	0.9683
Dataset 2:	0.7523	0.7294	0.7398